
Trabajo Teórico

Deep Learning en Robótica Móvil

Carlos Abadía Cutillas

Jose Antonio Castejón García



18-1-2023

Robots móviles

Índice

1. Introducción	3
2. Fundamentos de Deep Learning	4
3. Aplicaciones de Deep Learning en robótica móvil	5
4. Reinforcement Learning	6
4.1. Simuladores	7
5. Navegación	8
6. Planificación de trayectorias	12
7. Futuro del Deep Learning y Reinforcement Learning en robótica móvil	14
8. Conclusiones	16
9. Referencias	16

1. Introducción

La robótica móvil se refiere al uso de robots móviles para realizar tareas en el mundo real. Estos robots pueden ser utilizados en una amplia variedad de aplicaciones, desde la exploración del espacio hasta la industria y la agricultura. Con el auge de la inteligencia artificial, el Deep Learning se ha convertido en una herramienta cada vez más importante en el campo de la robótica móvil.

El Deep Learning es un subconjunto de la inteligencia artificial que se basa en el uso de redes neuronales para analizar grandes cantidades de datos y aprender por sí mismas. Esto permite a los robots móviles procesar y comprender información de manera más eficiente, lo que les permite realizar tareas de manera más autónoma.

Esto se está utilizando en una variedad de aplicaciones en robótica móvil, como el reconocimiento de objetos y la navegación autónoma. Sin embargo, también existen desafíos y limitaciones en el uso del Deep Learning en la robótica móvil, como la necesidad de grandes cantidades de datos y el consumo de energía. A pesar de estos desafíos, se espera que el Deep Learning continúe desempeñando un papel importante en el futuro de la robótica móvil y se espera que se desarrolle aún más en el futuro.

Este trabajo explorará los fundamentos del Deep Learning, más en concreto en el Reinforcement Learning, y sus aplicaciones actuales en la robótica móvil y los desafíos y limitaciones actuales. También discutiremos las predicciones sobre el futuro del Deep Learning en la robótica móvil y reflexionaremos sobre su impacto en el campo.

2. Fundamentos de Deep Learning

Los fundamentos del Deep Learning se basan en el uso de redes neuronales para analizar grandes cantidades de datos y aprender por sí mismas. Las redes neuronales son una red de nodos y enlaces que se inspiran en la forma en que funciona el cerebro humano. Cada nodo representa una unidad de procesamiento y cada enlace representa una conexión entre dos nodos.

Para entrenar una red neuronal, se le proporciona un conjunto de datos etiquetados, lo que significa que se le proporciona un conjunto de datos con las respuestas correctas ya conocidas. La red utiliza estos datos para ajustar los pesos de las conexiones entre los nodos de manera que pueda realizar tareas específicas de manera más precisa.

El proceso de ajuste de pesos se realiza mediante el uso de un algoritmo de optimización, como el gradiente descendente. Este algoritmo funciona iterando a través de todos los datos de entrenamiento y ajustando los pesos de manera que minimice la función de pérdida. La función de pérdida mide cuán incorrectas son las predicciones de la red y se trata de minimizarla para hacer que las predicciones de la red sean lo más precisas posibles.

Una vez que se ha completado el proceso de entrenamiento, la red está lista para utilizarse. Se le proporcionan nuevos datos (que no se utilizaron durante el entrenamiento) y utiliza lo que ha aprendido para realizar tareas específicas, como el reconocimiento de objetos.

3. Aplicaciones de Deep Learning en robótica móvil

El Deep Learning se está utilizando en una variedad de aplicaciones en robótica móvil. Algunos ejemplos de cómo se está utilizando el Deep Learning en la robótica móvil incluyen: reconocimiento de objetos, navegación autónoma, control de movimiento, seguimiento de personas, interacción con el entorno o agilidad y estabilidad.

En cuanto al reconocimiento de objetos, las redes neuronales pueden ser entrenadas para reconocer objetos específicos en imágenes o videos, lo que puede ser útil en aplicaciones de robótica móvil como la inspección de líneas de producción o la exploración de áreas peligrosas.

Para la tarea de la navegación autónoma, las redes neuronales también pueden ser utilizadas para permitir que los robots móviles naveguen de manera autónoma. Esto puede incluir la detección de obstáculos y la planificación de rutas seguras.

Otro ejemplo sería el control de movimiento, en el que el Deep Learning también puede ser utilizado, en este caso para controlar el movimiento de los robots móviles. Por ejemplo, se pueden entrenar redes neuronales para controlar brazos robóticos de manera precisa.

Otra aplicación para la que puede ser usado el Deep Learning es en el seguimiento de personas. Donde las redes neuronales pueden ser entrenadas para detectar y seguir a personas en imágenes o videos. Esto puede ser útil en

aplicaciones de seguridad o en el seguimiento de trabajadores en un entorno industrial.

El Deep Learning también puede ser utilizado para permitir que los robots móviles interactúen de manera más natural con su entorno. Además de poder mejorar la agilidad y estabilidad de los robots móviles. Por ejemplo, se pueden entrenar redes neuronales para controlar el equilibrio y la postura de un robot móvil en terrenos irregulares o para ajustar su velocidad y dirección en tiempo real.

4. Reinforcement Learning

El aprendizaje por refuerzo es un tipo de aprendizaje automático que consiste en entrenar a un agente para que realice acciones en un entorno con el fin de maximizar una recompensa. En el contexto de un robot que camina, el agente sería el robot y el entorno sería el mundo físico en el que opera. El robot sería entrenado para realizar acciones, como mover las piernas o ajustar su equilibrio, con el fin de maximizar la recompensa que recibe por caminar con éxito. Esto puede hacerse mediante Deep Learning, en el que se utiliza una red neuronal para procesar las observaciones que el robot recibe de sus sensores y emitir las acciones apropiadas. El Deep Learning por refuerzo implica el uso de redes neuronales profundas, capaces de procesar grandes cantidades de datos, en el proceso de aprendizaje por refuerzo. Puede utilizarse para entrenar a un robot a caminar y realizar otras tareas complejas.

4.1. Simuladores

Coordinar el movimiento de la marcha de un robot y la destreza de sus manipuladores ha resultado ser un problema difícil en el campo de la robótica. En el mundo físico, puede resultar difícil para los robots sortear obstáculos aparentemente pequeños debido a factores como la fricción de la superficie, la flexibilidad estructural, la vibración, el retardo de los sensores y la escasa transformación de los actuadores. Los simuladores ayudan a probar los modelos en el mundo virtual y paliar estos problemas para luego aplicar las soluciones y modelos obtenidos en el mundo real de manera sencilla.

Algunos de los simuladores más utilizados hoy en día son:

1. **MuJoCo** (Simulador físico de robots)
2. **OpenAI Gym** (Escenarios diversos (por ejemplo, control de robots, Go, Cart-Pole))
3. **DeepMind Lab** (Escena de laberinto 3D (robot))
4. **TORCS** (Simulador de coches de carreras)
5. **SIGVerse** (Simulador de robots (por ejemplo, dinámica, percepción, comunicación))
6. **AI2-THOR** (Entorno de simulación similar a escenas del mundo real)

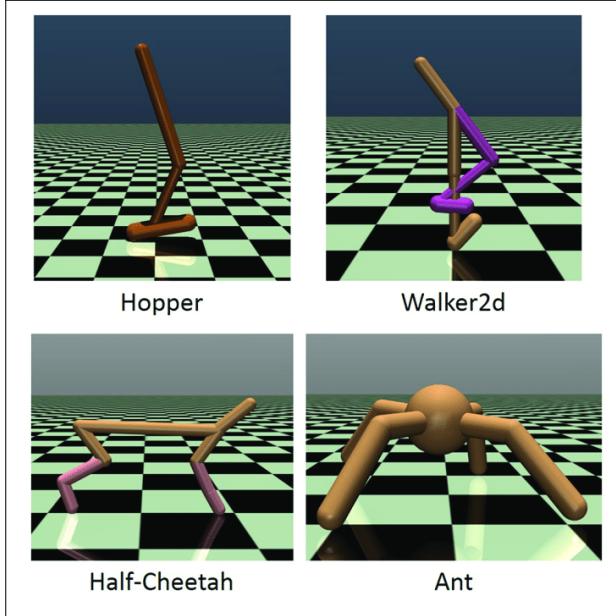


Figura 1: Simulador MuJoCo.

5. Navegación

En el ámbito de la navegación, se han desarrollado una serie de métodos de RL (Reinforcement Learning) que utilizan sensores como cámaras y radares para detectar obstáculos y construir mapas para la planificación de trayectorias.

Un método muy conocido es la localización y cartografía simultáneas, que construye mapas de forma incremental mediante la estimación de las posiciones en movimiento del robot. Sin embargo, los métodos tradicionales pueden resultar difíciles de utilizar en determinadas situaciones debido a sus exigencias computacionales y a su falta de adaptabilidad.

En los últimos años se han realizado varios estudios que han aplicado el RL a

la navegación de robots móviles tanto en entornos virtuales como en el mundo real. Algunos enfoques se han centrado en el entrenamiento de principio a fin de planificadores de movimiento sin mapa o arquitecturas neuronales diferenciables para permitir la navegación por trayectorias desconocidas. Otras investigaciones han utilizado el RL para mejorar la calidad de la evitación de obstáculos o para permitir a los robots aprender políticas de navegación punto a punto.

El RL también se ha aplicado al desarrollo de robots de servicio con capacidad de navegación y movimiento autónomos de largo alcance, lo que les permite transportar mercancías, equipajes y otros objetos a grandes distancias. Combinando el RL con algoritmos de control de la planificación del movimiento y arquitecturas de redes neuronales, estos robots pueden aprender a navegar con eficacia en entornos nuevos.

El RL también se ha utilizado para desarrollar comportamientos de navegación social en escenarios de interacción humano-robot, como la evasión de colisiones, las interacciones líder-seguidor y otros comportamientos. En aplicaciones prácticas, la eficacia, robustez, controlabilidad y precisión de los sistemas de navegación basados en RL se han probado y demostrado exhaustivamente. En general, el RL ha demostrado ser una potente herramienta para la navegación autónoma en robótica.

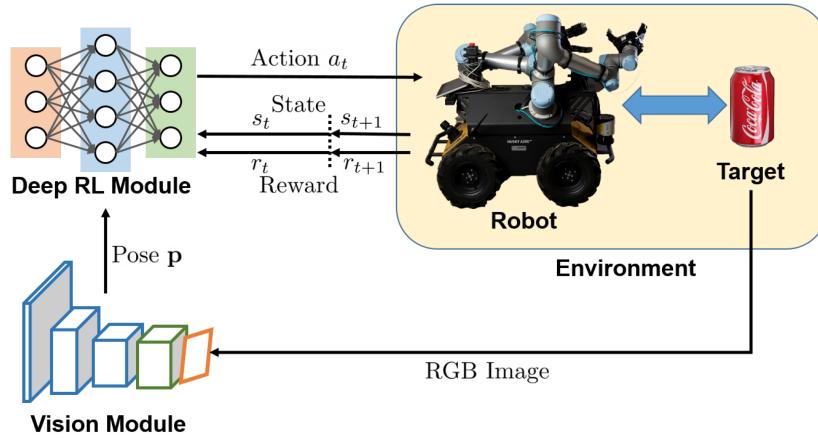


Figura 2: Sistema de navegación mediante RL.

Uno de los métodos más utilizados en este campo son los **Policy Gradient methods**.

Estos, son un tipo de algoritmo de aprendizaje por refuerzo que puede aplicarse a la navegación de robots móviles. Estos métodos funcionan optimizando directamente los parámetros de una política de red neuronal, utilizando el descenso de gradiente para buscar una política que maximice la señal de recompensa. La **política** es la regla a seguir por cada agente para decidir qué hacer en cada momento.

Uno de los métodos de **Policy Gradient** que se han aplicado a la navegación de robots móviles consiste en entrenar una red neuronal para predecir la acción óptima a realizar en cada paso temporal en función del estado actual del robot y de la ubicación del objetivo. La red neuronal se entrena utilizando una variante del descenso gradiente denominada **REINFORCE**, que permite a la red aprender una política que maximiza la recompensa esperada a lo largo del tiempo.

Otro enfoque consiste en utilizar métodos actor-crítico, que combinan una red de políticas (el 'actor') con una red de valores (el 'crítico'). La red política se entrena para predecir la acción óptima en cada paso temporal, mientras que la red de valor se utiliza para estimar la recompensa futura esperada para cada estado. La combinación de estas dos redes permite al robot aprender de forma más eficaz y tomar mejores decisiones sobre las acciones que debe emprender.

El algoritmo aplicado es el siguiente:

$$\nabla_{\theta} J(\theta) = E\tau \sim p\theta(\tau) [\sum_{t=0}^T \nabla_{\theta} \log \pi_{\theta}(a_t|s_t) R(\tau)] &= \sum_{t=0}^T \nabla_{\theta} \log \pi_{\theta}(a_t|s_t) \sum_{t'=t}^T r_{t'} \prod_{t''=t+1}^{t'} \gamma^{t''-t-1}$$

Donde:

1. θ representa los parámetros de la red política (el 'actor')
2. $J(\theta)$ es la recompensa acumulada esperada por la política
3. $\tau = (s_0, a_0, s_1, a_1, \dots, s_T)$ es una trayectoria de estados y acciones
4. $\pi_{\theta}(a_t|s_t)$ es la probabilidad de seleccionar la acción a_t en el estado s_t bajo la política actual
5. $R(\tau) = \sum_{t'=t}^T r_{t'} \prod_{t''=t+1}^{t'} \gamma^{t''-t-1}$ es la recompensa acumulada por la trayectoria, donde r_t es la recompensa en el paso t y γ es el factor de descuento

El algoritmo funciona iterativamente actualizando los parámetros de la red de políticas θ en la dirección del gradiente $\nabla_{\theta} J(\theta)$, con el fin de maximizar la recompensa acumulada esperada. Esto se suele hacer utilizando un algoritmo de optimización como el descentrado estocástico.

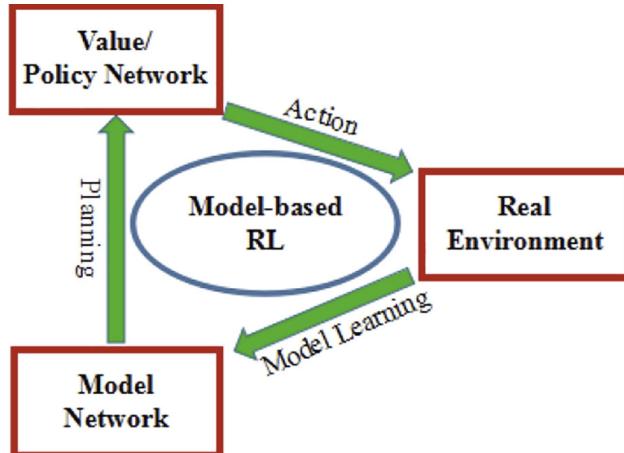


Figura 3: Método Policy Gradient.

En general, los métodos de gradiente de política han demostrado su eficacia en el entrenamiento de robots móviles para navegar por entornos complejos y realizar una amplia gama de tareas.

6. Planificación de trayectorias

En el ámbito de la planificación de trayectorias en robótica móvil, Los algoritmos tradicionales han tenido cierto éxito en este campo, pero a menudo adolecen de problemas como la baja precisión, la escasa estabilidad y la falta de solidez.

Así pues, se han desarrollado algoritmos multiagente descentralizados de evitación de colisiones para encontrar trayectorias libres de colisiones y eficientes en el tiempo, y se ha utilizado el aprendizaje de representación basado en datos para abordar problemas de planificación de trayectorias locales.

Por otro lado, las famosas redes neuronales convolucionales y los modos de

aprendizaje de extremo a extremo han sido fundamentales para la planificación de trayectorias en robótica, y se han utilizado para la planificación de trayectorias sin colisiones y eficientes en el tiempo en sistemas multirobot.

Sin embargo los que más están tomando fuerza y siendo utilizados hoy en día por su eficacia, son los algoritmos de **Q-Learning**. Los cuales se han aplicado para paliar los problemas de los métodos previamente mencionados con cierto éxito, y se han perfeccionado para reducir los requisitos de almacenamiento y ámbito de búsqueda. También se han utilizado técnicas de procesamiento de imágenes y métodos de RL para planificar trayectorias óptimas para robots móviles.

El algoritmo de **Q-Learning** es el siguiente:

$$Q(s, a) = (1 - \alpha) * Q(s, a) + \alpha * (r + \gamma * \max(Q(s', a')))$$

Donde:

1. s es el estado actual del sistema
2. a es la acción actual tomada
3. α es la tasa de aprendizaje (un valor entre 0 y 1 que determina el peso de la nueva información)
4. r es la recompensa recibida después de tomar la acción a en el estado s
5. γ es el factor de descuento (un valor entre 0 y 1 que determina la importancia de las recompensas futuras)
6. s' es el nuevo estado del sistema después de tomar la acción a en el estado s

7. a' es la siguiente acción tomada en el estado s'

Esta ecuación actualiza el valor Q para el par estado-acción actual basándose en la recompensa recibida y en el valor Q máximo del siguiente par estado-acción. Ayuda al algoritmo a aprender las acciones más óptimas para cada estado.

7. Futuro del Deep Learning y Reinforcement Learning en robótica móvil

Como hemos visto, el Deep Learning, más en concreto el Reinforcement Learning se pueden aplicar de manera a robótica móvil. Sin embargo, existen varios retos que pueden dificultar su eficacia en este contexto, como la complejidad de las muestras, la alta dimensionalidad de los espacios de datos y los problemas de convergencia.

Una solución a estos retos es el uso del aprendizaje profundo por refuerzo de máxima entropía (Maximum Entropy DRL). El Maximum Entropy DRL generaliza el objetivo del Reinforcement Learning usando un término de entropía y su objetivo es aprender una política que maximice la recompensa acumulada descontada con la regularización de entropía. Este enfoque permite la construcción de estrategias jerárquicas que pueden resolver eficazmente tareas complejas con recompensas dispersas a través del razonamiento probabilístico, al tiempo que elimina la necesidad de ajustar manualmente los hiperparámetros. En comparación con los métodos de búsqueda de políticas deterministas, **Maximum Entropy DRL** ha demostrado tener una mayor consistencia y robustez. La fórmula aplicada es la siguiente:

$$J(\pi) = \sum_{t=0}^{+\infty} E_{s_t, a_t \sim \rho_\pi} [r(s_t, a_t) + \alpha H(\pi(\cdot | s_t))] \quad (3)$$

Donde $H(\pi(\cdot | s_t))$ es la entropía de la política π en el estado s_t , y α es el hiperparámetro de temperatura que determina la importancia relativa del término de entropía.

Por otro lado, uno de los retos de la aplicación del aprendizaje por refuerzo (RL) a la robótica es el diseño de funciones de recompensa adecuadas y la exploración del tiempo. En el pasado, los investigadores recurrián a especificar manualmente las funciones de recompensa para que los algoritmos de RL las optimizaran.

Sin embargo, esto puede ser un factor limitante para los robots que necesitan comprender tareas y realizarlas en el mundo real. Para resolver este problema, algunos investigadores han tratado de capacitar a los robots para comprender conceptos semánticos y completar tareas combinando una pequeña cantidad de datos anotados con métodos de RL.

Esto ha llevado al desarrollo de robots que pueden aprender de demostraciones humanas, imitar y aprender acciones humanas y entender categorías semánticas como juguetes y bolígrafos. Estos enfoques pueden ser útiles para que los robots actúen con mayor eficacia en el mundo real.

8. Conclusiones

En conclusión, el Deep Learning tiene el potencial de tener un gran impacto en la robótica móvil en el futuro. El Deep Learning se está utilizando actualmente en una variedad de aplicaciones en la robótica móvil, como el reconocimiento de objetos, la navegación autónoma y el control de movimiento. Sin embargo, también existen algunos desafíos y limitaciones en el uso del Deep Learning en la robótica móvil, como la necesidad de grandes cantidades de datos etiquetados y el consumo de energía. En el futuro, se espera que el Deep Learning permita que los robots móviles se vuelvan más autónomos, procesen grandes cantidades de datos de manera más rápida y eficiente, se adapten mejor a nuevos entornos y tareas, colaboren más efectivamente con humanos y se integren más en la vida cotidiana de las personas.

9. Referencias

- **Vídeo de YouTube sobre Reinforcement Learning:**
<https://www.youtube.com/watch?v=bBnhdj9IqFg>
- **Reinforcement learning for robot research: A comprehensive review and open issues:**
<https://journals.sagepub.com/doi/10.1177/17298814211007305>
- **Towards Cognitive Exploration through Deep Reinforcement Learning for Mobile Robots:**
<https://arxiv.org/abs/1610.01733>
- **Deep Reinforcement Learning Based Mobile Robot Navigation: A Review:**
<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9409758>



- A Survey of Deep Learning Techniques for Mobile Robot Applications:

<https://arxiv.org/pdf/1803.07608.pdf>